

IN THE CLAIMS

1. (Previously Presented) In a vendor order server, a method for processing order messages, comprising the steps of:

receiving a first message of the order messages over a network, the first message comprising a first extended markup language document from a customer ordering application organized in a first predefined format;

directing the first message to a first message processing module of a plurality of message processing modules, directing the first message further comprising parsing the first message to determine a message type that identifies an ordering function for the first message, and directing the first message to the first processing module based on the message type;

obtaining a first data set from a first predefined element of the first extended markup language document based on the first predefined format of the first document in response to the step of receiving the first message;

obtaining a second data set by processing the first data set of the first message in response to the step of obtaining the first data set, further comprising invoking an ordering function based on a message type defined in a second predefined element of the first extended markup language document to generate the second data set; and

providing over the network the second data set in a second message comprising providing the second data set in a third predefined element in a second extended markup language document to the customer ordering application, the second extended markup language document organized in a second predefined format suitable for use by the customer ordering application.

Claims 2-4. (Cancelled)

-3-

5. (Original) The method of claim 1, wherein the step of obtaining the second data set comprises interacting with an order database based on the first data set and based on a message type of the first message to generate the second data set.
6. (Original) The method of claim 1, wherein the step of obtaining the second data set comprises performing an ordering function based on the first data set and based on a message type of the first message to generate the second data set.
7. (Original) The method of claim 1, wherein the second predefined format is suitable for integration into a database maintained by the ordering application.
8. (Cancelled)
9. (Previously Presented) A vendor order server for processing order messages, the order server comprising:
 - a memory;
 - an input/output interface in communication with the memory; and
 - a processor in communication with the memory and the input/output interface, wherein the memory is encoded with logic instructions for an order message manager application that, when performed on the processor, cause the processor to form an order message manager that processes order messages by performing the operations of:
 - receiving through the input/output interface a first message of the order messages over a network, the first message comprising a first extended markup language document from a customer ordering application organized in a first predefined format;

-4-

obtaining a first data set from a first predefined element of the first extended markup language document, the first message based on the first predefined format of the first extended markup language document in response to the step of receiving the first message;

obtaining a second data set by processing the first data set of the first message in response to the step of obtaining the first data set, further comprising invoking an ordering function based on a message type defined in a second predefined element of the first extended markup language document to generate the second data set; and

providing through the input/output interface over the network the second data set in a second message comprising providing the second data set in a third predefined element in a second extended markup language document to the customer ordering application, the second extended markup language document organized in a second predefined format suitable for use by an ordering application, the first and second predefined formats corresponding to the same order being processed by the order server.

10. (Cancelled)
11. (Original) The order server of claim 9, wherein the logic instructions for the order message manager application comprise further logic instructions, that, when performed on the processor, cause the order message manager to perform the operation of directing the first message to a first message processing module of a plurality of message processing modules.

12. (Previously Presented) The order server of claim 11, wherein the logic instructions for the order message manager application comprise further logic instructions, that, when performed on the processor, cause the order message manager to perform the operation of parsing the first message to determine a message type that identifies an ordering function for the first message, and directing the first message to the first processing module based on the message type, the message type indicative of the first and second predetermined formats.
13. (Original) The order server of claim 9, wherein the logic instructions for the order message manager application comprise further logic instructions, that, when performed on the processor, cause the order message manager to perform the operation of interacting with an order database based on the first data set and based on a message type of the first message to generate the second data set.
14. (Original) The order server of claim 9, wherein the logic instructions for the order message manager application comprise further logic instructions, that, when performed on the processor, cause the order message manager to perform the operation of performing an ordering function based on the first data set and based on a message type of the first message to generate the second data set.
15. (Original) The order server of claim 9, wherein the second predefined format is suitable for integration into a database maintained by the ordering application.

Claims 16-22. (Cancelled)

23. (Currently Amended) In a vendor order server, a method of processing order messages via an ordering network from a customer ordering application comprising:

defining an order message format having order data elements and order type elements;

defining an order response format corresponding to a response expected by the customer ordering application;

receiving, via the ordering network, an order message in the order message format, the order message having an order data set in the order ~~data elements~~~~message format~~ and an order type in the order type element;

parsing an order type from the order type element, the order type indicative of an ordering function performable by at least one of a plurality of order processing modules;

selectively invoking, based on the parsed order type, a particular one of the plurality of order processing modules corresponding to the order type, the invoking operable to route the order data set in the received order message to the invoked order processing module;

interacting with an order database coupled to the order server based on the message type, the order database responsive to the order processing module for transferring data corresponding to the parsed order type;

processing, based on the interacting, the routed order data set by the order processing module to generate an order response message in the order response format; and

transmitting the order response message to the ordering application, the order response format corresponding to an application database maintained by the ordering application, the defined order message format and order response format accessible to the ordering

-7-

application and the order server from predetermined types indicative of a predetermined format.

24. (Previously Presented) The method of claim 23 further comprising receiving the order response message by the ordering application and processing the order response message by employing the predetermined types in an automated manner that avoids operator intervention
25. (Previously Presented) The method of claim 24 wherein defining further comprises defining the order message format and order response format in a predetermined set of markup language definitions commonly accessible to both the order serve and the user application.
26. (Previously Presented) The method of claim 25 wherein receiving further comprises receiving the order response message in the order response message format recognizable via the commonly accessible markup definitions that define the predetermined format.
27. (Previously Presented) The method of claim 26 wherein processing further comprises processing in an automated manner that maps data items from the response message at the application database without manual reentry of data items via an interactive interface.
28. (Previously Presented) The method of claim 27 wherein the ordering application is conversant in XML and the order response format is an XML document type indicative of an XML element expected by the ordering application.

29. (Previously Presented) The method of claim 28 wherein identifying the message type involves parsing according to the predetermined format to identify tagged attributes indicative of the message types and processing by the order server further comprises indexing a mapping of order types to order processing modules.
30. (Previously Presented) The method of claim 29 wherein the order message format is an XML document definition and the order data elements and order type elements are XML tags in the XML document definition, wherein the order response format is operable for integration into an ordering database responsive to the order application.